View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Technical Disclosure Common

Technical Disclosure Commons

Defensive Publications Series

August 24, 2016

AUTOMATED METHOD OF NOISE REMOVAL FROM MULTICHANNEL AUDIO

Clayton Ritcher

Krzysztof Kulewski

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Ritcher, Clayton and Kulewski, Krzysztof, "AUTOMATED METHOD OF NOISE REMOVAL FROM MULTICHANNEL AUDIO", Technical Disclosure Commons, (August 24, 2016)
http://www.tdcommons.org/dpubs_series/255

This work is licensed under a Creative Commons Attribution 4.0 License.

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

AUTOMATED METHOD OF NOISE REMOVAL FROM MULTICHANNEL AUDIO

ABSTRACT

An automated system and method are disclosed for the removal of background noise from audio signals. The system takes input of multiple recordings of the same event and gives an output of a single audio stream. This audio stream emphasizes the common components of the inputs and attenuates components that are not perceived in all the recordings. The method removes constant background noise and shorter impulse noises effectively through , audio preprocessing, noise detection and attenuation, and final signal synthesis. Any video or audio editing software could use this method to create clearer audio signals with less background noise.

BACKGROUND

Audio recordings from video or pure audio typically contain unwanted background noise such as wind, nearby crowd chatter, or microphone bumps. This background noise is detrimental for playback experience because it makes the audio recording hard to hear. Therefore, removal of background noise is highly desirable.

Conventional techniques of automated noise removal include noise profiling and delay-and-sum beamforming. In noise profiling, a portion of the audio signal with very little of the desired audio (e.g. quiet before a song at a concert) is manually or automatically selected as a characterization of the background noise for the entire signal. This method is only effective at reducing noise, which is constant for the entire recording. Delay-and-sum beamforming synchronizes multiple audio streams to line up and simply averages the streams. This reinforces dominant audio sources that are heard in all the streams while diminishing noise not heard in all the streams. This method is effective for most types of noise, but requires many microphones for best results.

Conventional beamforming uses a microphone array with known spacing between each microphone and the relative time delays of each frequency component from one microphone to another to estimate the direction from which each frequency component is arriving at all times. Then, the frequency components are filtered based on their direction of arrival, allowing only sounds from a certain direction to pass through. This method of noise removal is often used by devices with multiple microphones for improved speech recognition and requires precise knowledge of the spacing between microphones.

The article titled "Frequency domain techniques for stereo to multichannel upmix" to Avendano et al. (proc. AES 22 Conference, 2002, pp. 1-10) explains a method that uses coherence between the stereo channels of a recording to extract ambient sound to be played out of the rear speakers in a surround sound setup. This method only attenuates noise that is uncorrelated across recordings. This method handles exactly two streams. Therefore, it is not utilized for multiple recordings.

Often, multiple audio recordings of the same event exist. Examples of events that are commonly recorded by multiple devices range from small events (e.g. kids playing in the backyard, flash mobs), to large events (e.g. concerts, press conferences, stage shows, and even to professionally recorded events such as sporting events). Thus, there is a need for an automated method to remove noise from multichannel audio.

DESCRIPTION

This disclosure presents an automated system and a method for removing background noise from multichannel audio. The automated system depicted in FIG. 1 takes in as input multiple recordings of the same event and gives an output of a single audio stream. This stream emphasizes the common components of the inputs and attenuates components that are not perceived in all the recordings. The method removes constant background noise and

shorter impulse noises effectively through three main processes viz., audio preprocessing, noise detection and attenuation, and final signal synthesis.

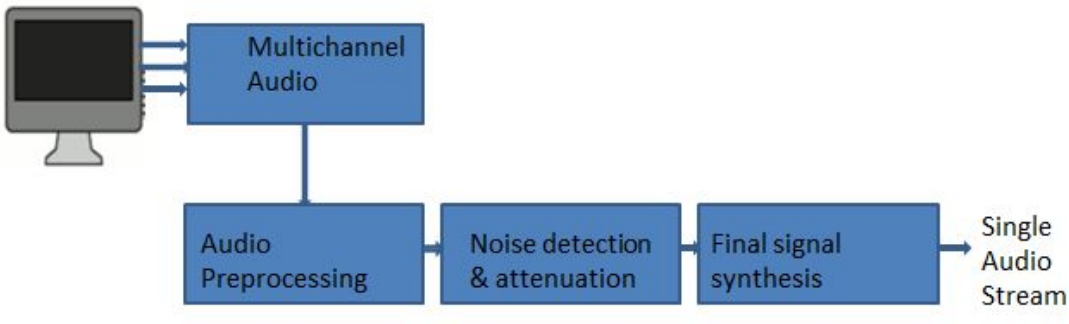

FIG. 1: Automated noise reduction system

Audio Preprocessing

Audio preprocessing of the input streams comprises the following steps:

1. Downmixing all audio signals from standard multi-channel to mono using a standard average of all the channels from each recording. Alternatively, any multichannel recording may be split by treating each channel as a separate recording.

2. Resampling all audio signals to be at the same sampling rate. A sampling rate of 11025 Hz is taken; however, other values could also be used. Each of these signals is called as $x_i(t)$, where $0 \leq i < n$, and n is the number of audio signals.

3. Computing the root mean square (RMS) of the concatenation of all audio signals to obtain a general idea of audio input loudness L.

4. Computing temporal offsets between audio signals using pairwise spectral cross correlation between audio streams and belief propagation for eliminating outliers.

5. Computing Short-time Fourier Transform (STFT) of each audio signal represented as STFTs $X_i(\tau,\omega)$, where τ is the synchronized time index synchronized using the time offsets

calculated above so that τ refers to the same real-world time for all of the STFTs, and ω is the frequency index. For the windowing function of the STFT, a Hamming window of size 5500 sample is used. The number of overlapping samples in each frame is considered to be 4500. A different type or size window and a different number of overlapping samples could also be used.

6. Assuming μ_i to be the mean of the magnitude of $X_i(\tau,\omega)$ and defining $X_i'(\tau,\omega) = X_i(\tau,\omega) / \mu_i$.

This centers the STFT around 1 and acts as a way of neglecting overall volume differences between recordings, making the method robust to overall volume differences between each recording.

Noise Detection and Attenuation

Noise detection and attenuation are performed by the following steps only for the available streams at each value of τ. Without loss of generality, we will continue as if all of the recordings are available at all times.

(i) We define a similarity metric between two STFTs at a single time–frequency point as

$s_{ij}(\tau,\omega) = 1 - (||X_i'(\tau,\omega)| - |X_j'(\tau,\omega)||) / (|X_i'(\tau,\omega)| + |X_j'(\tau,\omega)|)$.

If the denominator is 0, the similarity is set to 1 to avoid division by zero. This metric is essentially the percent similarity of the magnitudes of two corresponding time–frequency points. This similarity metric uses relative volume differences that correctly give the tone generated by the phone in the audience a low similarity across the two recordings because the tone was relatively much louder for one microphone than for the other. The graph in FIG 2 shows the similarity changes with the magnitudes of the points. The similarity ranges from 1 (when the magnitudes are both equal to 0) to 0 (when one magnitude is 0 while the other is not).

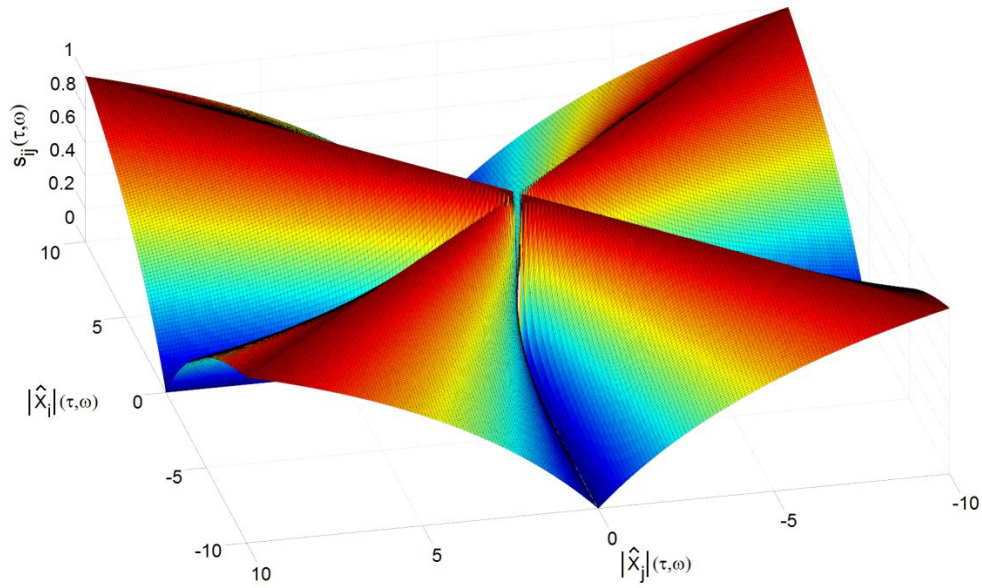

FIG. 2: Similarity graph

(ii) We map the similarity metric to a measure of the original signal allowed to pass through at a given time–frequency point. In general, this value is near zero for points with a very low similarity and near 1 for points with high similarity. The amount of original signal passed through is defined as p_ij(τ,ω) = (1/***π***) * arctan(k * (s_ij(τ,ω) - c)) + 0.5, where k determines the sharpness of the drop-off of the curve, and c specifies the occurrence of the cutoff point. FIG. 3 is shown using k = 50 and c = .5 while these parameters could also be changed.

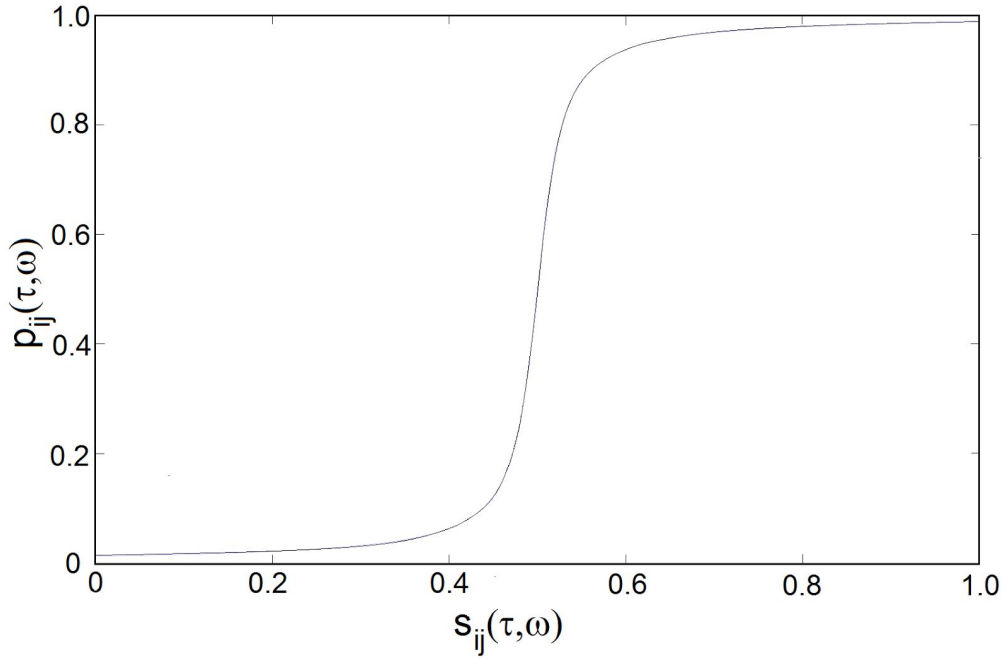

FIG. 3: Attenuation graph

(iii) We define Y_ij(τ,ω) = p_ij(τ,ω) * (X_i′(τ,ω) + X_j′(τ,ω)) / 2, which is the noise attenuated combination of the i^{th} and j^{th} STFTs at the given point.

Final Signal Synthesis

Final signal synthesis is carried out by the following steps:

A) Calculate the final combined STFT, Z(τ,ω), as the sum of each unique pairwise combination, by assuming there are n signals available at time τ. This is calculated as Z(τ,ω) = (2 / (n * (n1)) * sum from i = 0 to n – 2 { sum from j = i + 1 to n - 1{Y_ij(τ,ω) } }.

B) Use the overlap-add method to find the inverse STFT of Z(τ,ω), z(t). Same parameters i.e. window and overlap used to originally calculate the STFTs are used here.

C) Calculate the RMS of z(t), L_z, and find the final output signal, o(t) = z(t) * (L / L_z). This scales the output signal to have a similar volume level as the input signals.

Any video or audio editing software could use this method to create clearer media with less background noise. It could also be used to remove unwanted noise during

teleconferencing or speech recognition. It could be implemented at the hardware level in devices with multiple microphones for improved recordings. Prior knowledge of the positions of the microphones that recorded the audio is not required.